Package 'easyPubMed'

February 27, 2017

Type Package

Title Search and Retrieve Scientific Publication Records from PubMed

Version 2.3

Date 2017-02-26

Author Damiano Fantini

Maintainer Damiano Fantini <damiano.fantini@gmail.com>

Description Query NCBI Entrez and retrieve PubMed records in XML or text format. Process PubMed records by extracting and aggregating data from selected fields. A large number of records can be easily downloaded via this simple-to-use interface to the NCBI PubMed API.

Depends R(>= 3.1), XML

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-27 08:15:26

R topics documented:

easyPubMed-package *Retrieve and Process Scientific Publication Records from Pubmed*

Description

Query NCBI Entrez and retrieve PubMed records in XML or TXT format. PubMed records can be downloaded and saved as XML or text files. Data integrity is enforced during data download, allowing to retrieve and save very large number of records effortlessly. PubMed records can be processed to extract publication- and author-specific information.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
# Example 01: retrieve data in XML format
dami_query_string <- "Damiano Fantini[AU]"
dami_on_pubmed <- get_pubmed_ids(dami_query_string)
dami_papers <- fetch_pubmed_data(dami_on_pubmed)
titles <- unlist(xpathApply(dami_papers, "//ArticleTitle", saveXML))
title_pos <- regexpr('<ArticleTitle>.*<\\/ArticleTitle>', titles)
titles <- substr(titles, title_pos + 14, title_pos + attributes(title_pos)$match.length - 16)
print(titles)
#
# Example 02: retrieve data in TXT format
dami_query_string <- "Damiano Fantini[AU]"
dami_on_pubmed <- get_pubmed_ids(dami_query_string)
dami_papers <- fetch_pubmed_data(dami_on_pubmed, format = "abstract")
dami_papers[dami_papers == ""] <- "\n"
cat(paste(dami_papers[1:65], collapse = ""))
#
## Not run:
# Example 03: retrieve data from PubMed and save as XML file
ml_query <- "Machine Learning[TI] AND 2016[PD]"
out1 <- batch_pubmed_download(pubmed_query_string = ml_query, batch_size = 180)
XML::xmlParse(out1[1])
#
# Example 04: retrieve data from PubMed and save as TXT file
ml_query <- "Machine Learning[TI] AND 2016[PD]"
out2 <- batch_pubmed_download(pubmed_query_string = ml_query, batch_size = 180, format = "medline")
readLines(out2[1])[1:30]
#
# Example 05: extract information from a single PubMed record
ml_query <- "Machine Learning[TI] AND 2016[PD]"
out3 <- batch_pubmed_download(pubmed_query_string = ml_query, batch_size = 180)
```

```
PM_data <- articles_to_list(out3[1])
PM_record_df <- article_to_df(PM_data[[100]])
print(PM_record_df[1,])
print(PM_record_df[,"address"])
#
# Example 06: query PubMed and extract information from multiple records in one step
ml_query <- "Machine Learning[TI] AND 2016[PD]"
out4 <- batch_pubmed_download(pubmed_query_string = ml_query, batch_size = 180)
PM_tab <- table_articles_byAuth(out4[1], autofill = TRUE, included_authors = "last")
PM_tab$address <- substr(PM_tab$address, 1, 15)
PM_tab[50:70,c("pmid", "jabbrv", "year", "lastname", "address")]
#

## End(Not run)
```

`articles_to_list` *Cast PubMed Data into a List of Articles*

Description

Convert an XML object of PubMed records into a list of strings (character vector of length 1) corresponding to individual PubMed articles. PubMed records are identified by a "/PubmedArticle" XML tag. This automatically cast all the content of each PubMed record to a character-class object without removing XML tags.

Usage

```
articles_to_list(pubmed_data)
```

Arguments

`pubmed_data`	String corresponding to the name of an XML file (typically, the result of a batch_pubmed_download() call). Alternatively, an XML Object, such as the result of a fetch_pubmed_data() call.

Details

The input is an XML object or an XML file, typically the result of a fetch_pubmed_data() call or a batch_pubmed_download() call. The function returns a list where each element is a different PubMed record.

Value

List of character elements including the all records from the original XML object. Elements in the list are not named and are only accessible via the numeric index.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
#
# retrieve PubMed data and return a list ot articles
dami_query <- "Damiano Fantini[AU]"
outfile <- batch_pubmed_download(dami_query, dest_file_prefix = "easyPM_ex001_")
listed_articles <- articles_to_list(pubmed_data = outfile)
listed_articles[[3]]
```

---

article_to_df *Extract Data from a PubMed Record*

Description

Extract publication-specific information from a PubMed record driven by XML tags. The input record is a string (character-class vector of length 1) and includes PubMed-specific XML tags. Data are returned as a data frame where each row corresponds to one of the authors of the PubMed article.

Usage

```
article_to_df(pubmedArticle, autofill = FALSE, max_chars = 500)
```

Arguments

pubmedArticle	String including one PubMed record.
autofill	Logical. If TRUE, missing affiliations are automatically imputed based on other non-NA addresses from the same record.
max_chars	Numeric (>=0). Maximum number of characters to be extracted from the Article Abstract field.

Details

Given one Pubmed Article record, this function will automatically extract a set of features. Extracted information include: PMID, DOI, article title, article abstract, publication date (year, month, day), journal name (title, abbreviation) and a set of author-specific info (names, affiliation, email address). Each row of the output data frame corresponds to one of the authors of the PubMed record. Author-independent info (publication ID, title, journal, date) are identical across all rows.

Value

Data frame including the extracted features. Each row correspond a different author.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
#
# Query PubMed, retrieve a selected citation and format it as a data frame
dami_query <- "Damiano Fantini[AU]"
dami_on_pubmed <- get_pubmed_ids(dami_query)
dami_abstracts_xml <- fetch_pubmed_data(dami_on_pubmed)
dami_abstracts_list <- articles_to_list(dami_abstracts_xml)
article_to_df(pubmedArticle = dami_abstracts_list[[4]], autofill = FALSE, max_chars = 100)
article_to_df(pubmedArticle = dami_abstracts_list[[4]], autofill = TRUE, max_chars = 300)[1:2,]
```

---

batch_pubmed_download *Download PubMed Records in XML or TXT Format*

Description

Performs a PubMed Query (via the get_pubmed_ids() function), downloads the resulting data (via multiple fetch_pubmed_data() calls) and then saves data in a series of xml or txt files on the local drive. The function is suitable for downloading a very large number of records.

Usage

```
batch_pubmed_download(pubmed_query_string, dest_dir = NULL,
                      dest_file_prefix = "easyPubMed_data_", format = "xml",
                      batch_size = 400, res_cn = 1)
```

Arguments

`pubmed_query_string`
: String (character-vector of length 1): this is the string used for querying PubMed (the standard PubMed Query synthax applies).

`dest_dir`
: String (character-vector of length 1): this string corresponds to the name of the existing folder where files will be saved. Existing files will be overwritten. If NULL, the current working directory will be used.

`dest_file_prefix`
: String (character-vector of length 1): this string is used as prefix for the files that are written locally.

`format`
: String (character-vector of length 1): data will be requested from Entrez in this format. Acceptable values are: c("medline","uilist","abstract","asn.1", "xml") When format != "xml", data will be saved as text files (txt).

batch_size	Integer (1 < batch_size < 5000): maximum number of records to be saved in a single xml or txt file.
res_cn	Integer (> 0): numeric index of the data batch to start downloading from. This parameter is useful to resume an incomplete download job after a system crash.

Details

Download large number of PubMed records as a set of xml or txt files that are saved in the folder specified by the user. This function enforces data integrity. If a batch of downloaded data is corrupted, it is discarded and downloaded again. Each download cycle is monitored until the download job is successfully completed. This function should allow to download a whole copy of PubMed, if desired. The function informs the user about the current progress by constantly printing to console the number of batches still in queue for download. pubmed_query_string accepts standard PubMed synthax. The function will query PubMed multiple times using the same query string. Therefore, it is recommended to use a [EDAT] or a [PDAT] filter in the query if you want to ensure reproducible results.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
## Not run:
# Example 01: retrieve data from PubMed and save as XML file
ml_query <- "Machine Learning[TI] AND 2016[PD]"
out1 <- batch_pubmed_download(pubmed_query_string = ml_query, batch_size = 180)
XML::xmlParse(out1[1])
#
# Example 02: retrieve data from PubMed and save as TXT file
ml_query <- "Machine Learning[TI] AND 2016[PD]"
out2 <- batch_pubmed_download(pubmed_query_string = ml_query, batch_size = 180, format = "medline")
readLines(out2[1])[1:30]

## End(Not run)
```

---

custom_grep *Retrieve Text Between XML Tags*

Description

Extract text form a string containing XML or HTML tags. Text included between tags of interest will be returned. If multiple tagged substrings are found, they will be returned as different elements of a list or character vector.

Usage

```
custom_grep(xml_data, tag, format = "list")
```

Arguments

xml_data	String (of class character and length 1): corresponds to the PubMed record or any string including XML/HTML tags.
tag	String (of class character and length 1): the tag of interest (does NOT include < > chars).
format	c("list", "char"): specifies the format for the output.

Details

The input string has to be a character string (length 1) containing tags (HTML or XML format). If an XML Document is provided as input, the function will rise an error.

Value

List or vector where each element corresponds to an in-tag substring.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
#
# extract substrings based on regular expressions
string_01 <- "I can't wait to watch the <strong>Late Night Show with"
string_01 <- paste(string_01, "Seth Meyers</strong> tonight at <strong>11:30</strong>pm CT!")
custom_grep(xml_data = string_01, tag = "strong", format = "char")
custom_grep(xml_data = string_01, tag = "strong", format = "list")
```

---

fetch_pubmed_data *Retrieve PubMed Data in XML or TXT Format*

Description

Retrieve PubMed records from Entrez following a search performed via the get_pubmed_ids() function. Data are downloaded in the XML or TXT format and are retrieved in batches of up to 5000 records.

Usage

```
fetch_pubmed_data(pubmed_id_list, retstart = 0, retmax = 500, format = "xml")
```

Arguments

pubmed_id_list	List: the result of a get_pubmed_ids() call.
retstart	Integer (>=0): index of the first UID in the retrieved PubMed Search Result set to be included in the output (default=0, corresponding to the first record of the entire set).
retmax	Integer (>=1): size of the batch of PubMed records to be retrieved at one time.
format	Character: element specifying the output format. The following values are allowed: c("asn.1", "xml", "medline", "uilist", "abstract").

Details

Retrieve PubMed records based on the results of a get_pubmed_ids() query. Records are retrieved from Entrez via the PubMed API efetch function. The first entry to be retrieved may be adjusted via the retastart parameter (this allows the user to download large batches of PubMed data). The maximum number of entries to be retrieved can also be set adjusting the retmax parameter (1 < retmax < 5000). Data will be downloaded on the fly (no files are saved locally as a result of a fetch_pubmed_data() call).

Value

If format == "xml": a XMLInternalDocument-class object. For accessing these data, use a XML parser. If format != "xml": a "character" vector. Each element corresponds to one line of data.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/
https://www.ncbi.nlm.nih.gov/books/NBK25499/table/chapter4.T._valid_values_of__retmode_and/

Examples

```
## Not run:
# Example 01: retrieve data in XML format
dami_query_string <- "Damiano Fantini[AU]"
dami_on_pubmed <- get_pubmed_ids(dami_query_string)
dami_papers <- fetch_pubmed_data(dami_on_pubmed)
titles <- unlist(xpathApply(dami_papers, "//ArticleTitle", saveXML))
title_pos <- regexpr("<ArticleTitle>.*<\\/ArticleTitle>", titles)
titles <- substr(titles, title_pos + 14, title_pos + attributes(title_pos)$match.length - 16)
print(titles)
#
```

```
## End(Not run)
# Example 02: retrieve data in TXT format
dami_query_string <- "Damiano Fantini[AU]"
dami_on_pubmed <- get_pubmed_ids(dami_query_string)
dami_papers <- fetch_pubmed_data(dami_on_pubmed, format = "abstract")
dami_papers[dami_papers == ""] <- "\n"
cat(paste(dami_papers[1:65], collapse = ""))
```

`get_pubmed_ids` *Simple PubMed Record Search*

Description

Query PubMed (Entrez) in a simple way via the PubMed API eSearch function. Calling this function results in posting the results on the PubMed History Server. This allows later access to the resulting data via the fetch_pubmed_data() function.

Usage

```
get_pubmed_ids(pubmed_query_string)
```

Arguments

`pubmed_query_string`
: is a character vector and is the String that is used for querying PubMed (standard PubMed synthax, see reference for details).

Details

This function will use the String provided as argument for querying PubMed via the eSearch function of the PubMed API. The Query Term can include one or multiple words, as well as the standard PubMed operators (AND, OR, NOT) and tags (i.e., [AU], [PDAT], [Affiliation], and so on). ESearch will post the UIDs resulting from the search operation onto the History server so that they can be used directly in a subsequent fetchPubmedData() call.

Value

The function returns a list. The list includes the number of records found on PubMed and the first 20 PubMed IDs (UID) retrieved by the query. The list also includes QueryKey and WebEnv that are required for a subsequent fetch_pubmed_data() call.

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/
https://www.ncbi.nlm.nih.gov/books/NBK3827/#_pubmedhelp_Search_Field_Descriptions_and_

Examples

```
##  Search for scientific articles written by Damiano Fantini
##  and print the number of retrieved records to screen.
##  Also print the retrieved UIDs to screen.
##
dami_on_pubmed <- get_pubmed_ids("Damiano Fantini[AU]")
print(dami_on_pubmed$Count)
print(unlist(dami_on_pubmed$IdList))
```

---

table_articles_byAuth *Extract Publication and Affiliation Data from PubMed Records*

Description

Extract Publication Info from PubMed records and cast data into a data.frame where each row corresponds to a different author. It is possible to retrieve data from first authors or last authors only as well as information from all authors of each PubMed record.

Usage

```
table_articles_byAuth(pubmed_data,
                      included_authors = "all",
                      max_chars = 500,
                      autofill = TRUE,
                      dest_file = NULL)
```

Arguments

pubmed_data	PubMed Data in XML format: typically, an XML file resulting from a batch_pubmed_download() call or an XML object, result of a fetch_pubmed_data() call.
included_authors	Character: c("first", "last", "all"). Only includes information from the first, the last or all authors of a PubMed record.
max_chars	Numeric: maximum number of chars to extract from the AbstractText field.
autofill	Logical. If TRUE, missing affiliations are imputed according to the available values (from the same article)
dest_file	String (character of length 1). Name of the file that will be written for storing the output. If NULL, no file will be saved.

Details

Retrieve publiccation and author information from PubMed data in the form of a data frame.

Value

Data frame including the following fields: c("article.title","article.abstract", "date.year","date.month","date.day", "journal.abbrv","journal.title", "auth.last","auth.fore","auth.address", "auth.email").

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
## Not run:
#
#
dami_query <- "Damiano Fantini[AU]"
dami_on_pubmed <- get_pubmed_ids(dami_query)
dami_abstracts_xml <- fetch_pubmed_data(dami_on_pubmed)
table_articles_byAuth(pubmed_data = dami_abstracts_xml,
                      included_authors = "first",
                      max_chars = 100,
                      autofill = TRUE)[1:2,]
#
#
dami_query <- "Damiano Fantini[AU]"
curr.file <- batch_pubmed_download(dami_query, dest_file_prefix = "test_bpd_")
table_articles_byAuth(pubmed_data = curr.file[1],
                      included_authors = "all",
                      max_chars = 20,
                      autofill =FALSE)

## End(Not run)
```

---

trim_address *Trim and Format Address Information*

Description

Set of rules for trimming and standardizing the format of address information retrieved from PubMed records. Affiliations including more than one address will be trimmend and only the first address will be returned.

Usage

```
trim_address(addr)
```

Arguments

addr	Character string including an address as extracted from PubMed records.

Value

Character string including a formatted and trimmed address (if available).

Author(s)

Damiano Fantini <"damiano.fantini@gmail.com">

References

http://www.biotechworld.it/bioinf/2016/01/05/querying-pubmed-via-the-easypubmed-package-in-r/

Examples

```
addr_string <- " 2 Dept of Urology, Feinberg School of Medicine,"
addr_string <- paste(addr_string, "Chicago, US; Dept of Mol Bio as well...")
addr_string
trim_address(addr = addr_string)
```

Index